Section 2  
RAPPORT DU PROJET II :

# Gestion des inscriptions en ligne pour la formation KFOKAM 48

## 

# INTRODUCTION

## Contexte

L’école de formation KFOKAM48 accueille chaque année des centaines d’étudiants. L’inscription de ces étudiants ainsi que leur répartition dans les différentes formations n’est pas une chose aisée à gérer pour le personnel de la scolarité. Le service de scolarité doit collecter une multitude de fiches remplies par les étudiants et les classer manuellement. Il a également la lourde tâche d’attribuer des numéros de matricules à tous les étudiants inscrits. L’absence d’une application informatique permettant la gestion en ligne entraîne un travail pénible pour les agents du service de la scolarité qui se voient submergés et une longue file d’attente pour les étudiants.

## Objectifs

Notre projet vise à mettre sur pied une base de données qui va faciliter l'enrôlement de nouveaux étudiants à travers un système intelligent de « **Gestion des Inscriptions en Ligne ».** Le système doit permettre d’enregistrer et de suivre les étudiants, les formations, les inscriptions, et les paiements.

# MÉTHODOLOGIE

Pour parvenir à un système qui répond pleinement aux exigences susmentionnées, nous avons choisi de procéder méthodiquement.

## Choix des entités et définition des associations

Suivant les exigences de notre système qui sont entre autres :

1. ***Pour les étudiants :***

* Création du compte étudiant ;
* **Consultation des offres de formation ;**
* **Inscription en ligne ;**
* **Suivi de l'inscription ;**
* **Gestion du compte ;**

1. ***Pour l’administration de l’école :***

* ***La gestion des inscriptions***
* Enregistrement des inscriptions ;
* Dénombrement des inscrits ;
* Recherche de inscrits par filière;

1. ***La gestion des formations :***

* Création, modification et suppression des offres de formation ;
* Affichage des formations offertes ainsi que le nombre de place disponible ;
* Appréciation des formations les plus sollicitées ;
* Définition les prérequis et les capacités d'accueil pour chaque formation ;
* Fixation des dates d'ouverture et de clôture des inscriptions.

1. **la gestion des paiements :**

* Effectuer le paiement en ligne
* lister les paiements effectués par les étudiants
* Émettre des factures et des reçus.

Nous avons répertorié 4 entités et 3 associations pour remplir ces exigences. Il s’agit de :

**Entités**

* **Formations** : id\_formation, nom\_formation, description, date\_debut, date\_fin, tarif
* **Etudiants** : id\_etudiant, nom, prenom, date\_naissance, email, telephone, date\_inscription
* **Inscriptions** : id\_inscription, id\_etudiant, id\_formation, date\_inscription, statut\_paiement
* **Paiements** : id\_paiement, id\_inscription, montant, date\_paiement, methode\_paiement

**Associations**

| **Association** | **Cardinalité** | **Explication** |
| --- | --- | --- |
| Prendre en compte | (1.1 : 1.N) | 1 inscription prend en compte une et une seule formation ; et une formation est prise en compte par une ou plusieurs informations. |
| Faire | (1.1 : 1.N) | 1 inscription peut être faite par 1 et 1 seul étudiant tandis qu’un étudiant peut faire une ou plusieurs inscriptions. |
| Demande | (1:N : 1.1) | 1 inscription peut se faire en un ou plusieurs paiements et un paiement correspond à une et une seule inscription. |

## Création de la Base de données et des tables

Suivant le rendu de notre MCD présenté dans la documentation technique, nous avons créé une base de données **“ecole ”** ; des tables à base du langage SQL et nous leur avons attribué les caractéristiques suivantes.

1. **Table Formations**

| **Caractéristiques** | **Nom du champ** | **Type** | **NULL?** |
| --- | --- | --- | --- |
| Identifiant de la formation | id\_formation | AUTO\_INCREMENT | Non |
| Nom de la formation | nom\_formation | VARCHAR (50) | Non |
| Description de la formation | description | TEXT | Oui |
| Date de début de la formation | date\_debut | DATE | Non |
| Date de fin de la formation | date\_fin | DATE | Non |
| Coût de la formation | tarif | Double unsigned | Non |
| nombre de places disponibles | nombre\_places | INT(50) | Non |

1. **Table Etudiants**

| **Caractéristiques** | **Nom du champ** | **Type** | **NULL?** |
| --- | --- | --- | --- |
| Identifiant de l’étudiant | id\_etudiant | AUTO\_INCREMENT | Non |
| Nom de l’étudiant | nom | VARCHAR (50) | Non |
| Prénom de l’étudiant | prénom | VARCHAR (50) | non |
| Date de naissance | Date\_naissance | DATE | non |
| Adresse email de l’étudiant | email | VARCHAR (50) | Non |
| Numéro de téléphone de l’étudiant | telephone | VARCHAR (50) | Non |
| Date d’ouverture de compte | date\_inscription | DATE | Non |

1. **Table Inscriptions**

| **Caractéristiques** | **Nom du champ** | **Type** | **NULL?** |
| --- | --- | --- | --- |
| Identifiant de l’inscription | id\_inscription | AUTO\_INCREMENT | Non |
| Date de l’inscription à une formation | date\_inscription | DATE | Non |
| Etat de solvabilité d’un étudiant | Statut paiement | tinyint(1) | Non |
| Clé étrangère de la table etudiant | id\_etudiant\_Etudiants | int | Non |
| Clé étrangère de la table formation | id\_formation\_formations | int | Non |

1. **Table Paiements**

| **Caractéristiques** | **Nom du champ** | **Type** | **NULL?** |
| --- | --- | --- | --- |
| L’identifiant du paiement | id\_paiement | AUTO\_INCREMENT | Non |
| Montant versé par l’étudiant | montant | double | Non |
| Date de paiement | date\_paiement | date | Non |
| Méthode de paiement | methode\_paiement | VARCHAR (50) | Non |
| Clé étrangère de la table inscription | id\_inscription\_Inscriptions | int | Non |

## 3. Ajout des données

La création de nos différentes tables étant faite, nous avons pour une opérationnalisation de notre base de données inséré une série de données pour tester son bon fonctionnement.

Chacune de ces données remplit pleinement les caractéristiques que nous avons définies dans les tables ci-dessus. Nous avons respectivement dans notre bibliothèque à l’heure actuelle :

* 06 Formations ;
* 09 potentiels étudiants ;
* 09 inscrits ;
* 17 paiements dont 7 par carte bancaire, 6 par chèque et 4 par orange money.

## 4. Création des requêtes

Notre système en plus de gérer les inscriptions aide le service de scolarité à émettre des rapports et à faire un bilan de gestion ceci à travers des requêtes qu’il envoie au système.

Nous avons prévu des requêtes pour :

* lister tous les étudiants inscrits à une certaine formation ;
* trouver les paiements effectués par un certain étudiant ;
* lister les inscriptions en cours d'un étudiant ;
* compter le nombre total d'inscriptions par formation.

## 5. Définition des fonctions et procédures de notre système

Une fonction est un code qui effectue une série d'instructions bien précises, et renvoie le résultat de ces instructions.

En matière de fonctions et de procédures, notre système prend en compte :

* Une procédure d’ajout d’une inscription
* Une fonction de vérification de la disponibilité d’une formation (nombre de places restantes).

## 6. Mise en place des vues et triggers

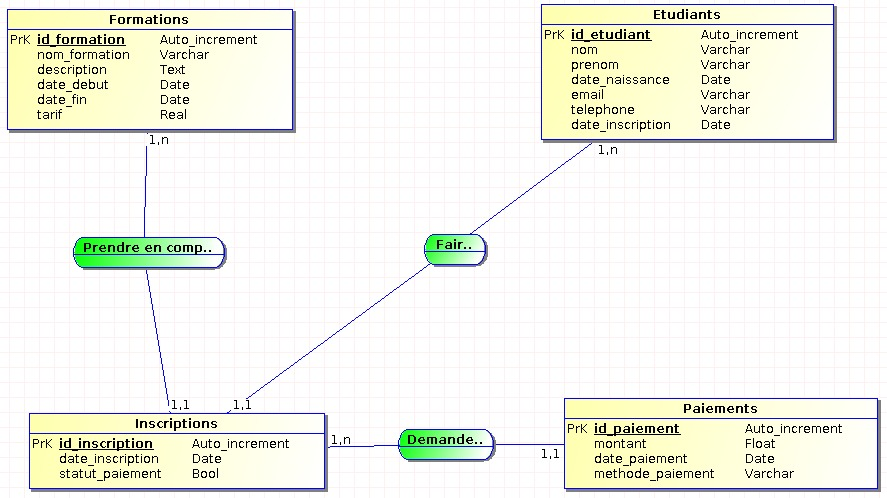
Les **vues** sont des objets de la base de données, constitués d'un nom, et d'une requête de sélection. Une fois qu'une vue est définie, on peut l'utiliser comme on le ferait avec une table ; table qui serait constituée des données sélectionnées par la requête définissant la vue. Pour notre projet nous avons défini une vue **pour afficher les détails des inscriptions** (nom de la formation, nom de l'étudiant, date d'inscription, statut de paiement)**.**

Les triggers (ou déclencheurs) sont des objets de la base de données. Attachés à une table, ils vont déclencher l'exécution d'une instruction, ou d'un bloc d'instructions, lorsqu'une, ou plusieurs lignes sont insérées, supprimées ou modifiées dans la table à laquelle ils sont attachés. Pour notre projet nous avons créez un trigger pour **mettre à jour le statut de paiement** lors de l'ajout d'un paiement

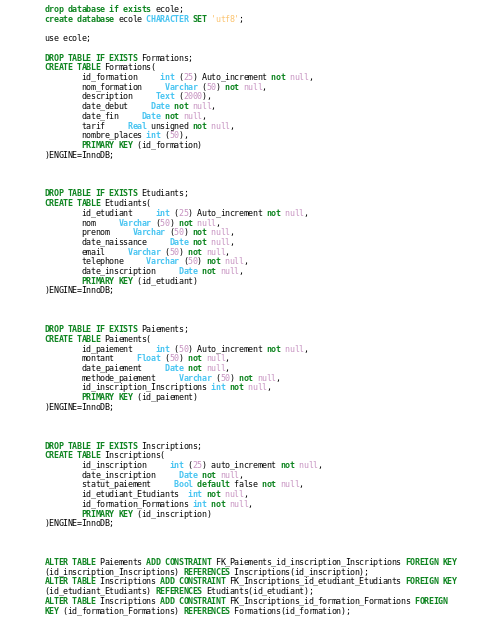
# Résultats

Dans cette section vous trouverez pour chaque étape de notre projet des captures d'écran montrant l’ensemble de nos actions sous forme de scripts SQL et le rendu lors de l’implémentation dans MySQL

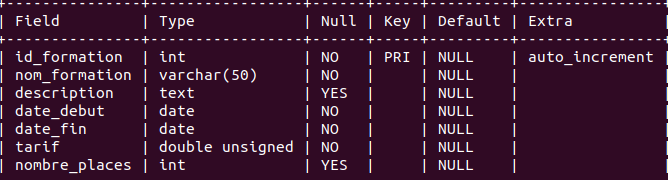
## Choix des entités et définition des associations

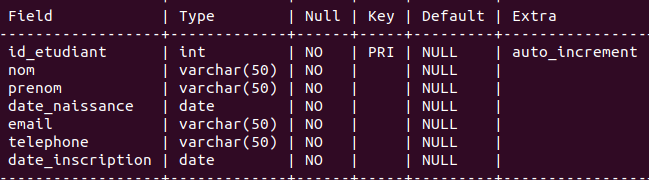


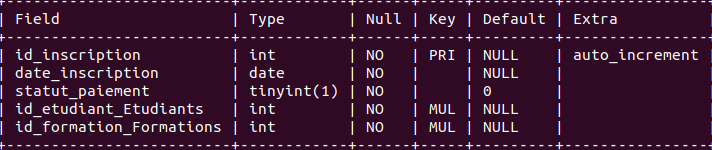
## Création de la base de données et des tables

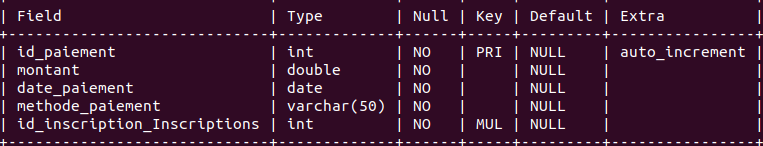


## Affichage des tables dans MySql





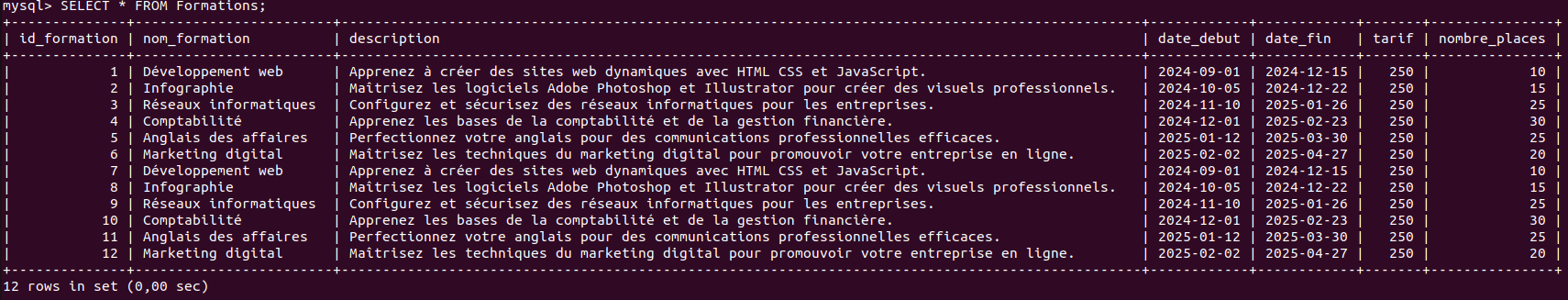


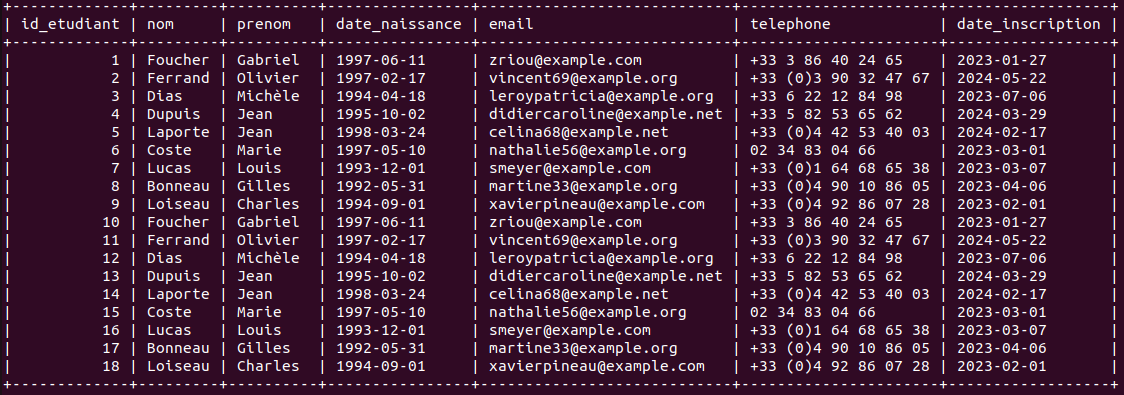


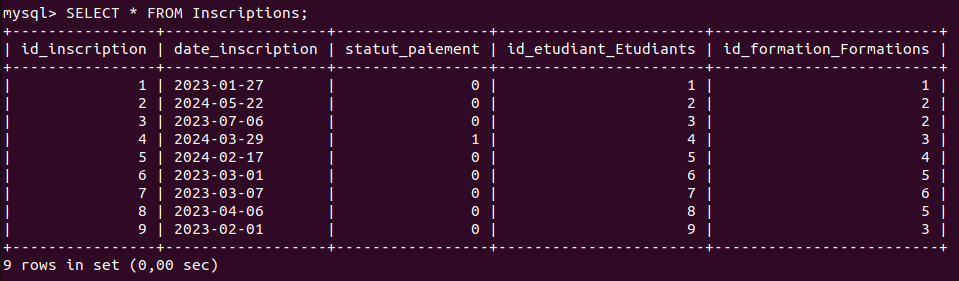
## Ajout des données

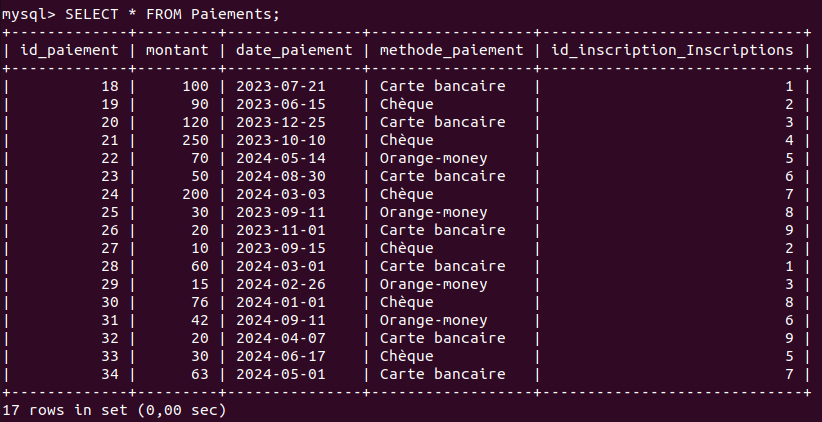


## Affichage des tables remplies dans MySql

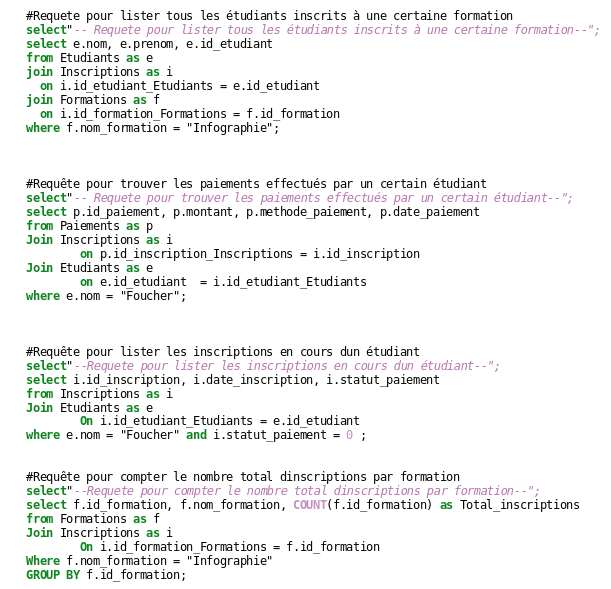






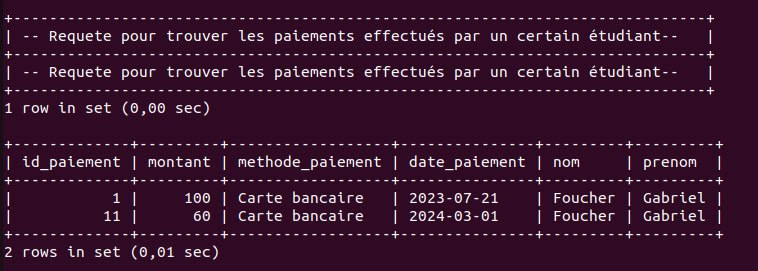


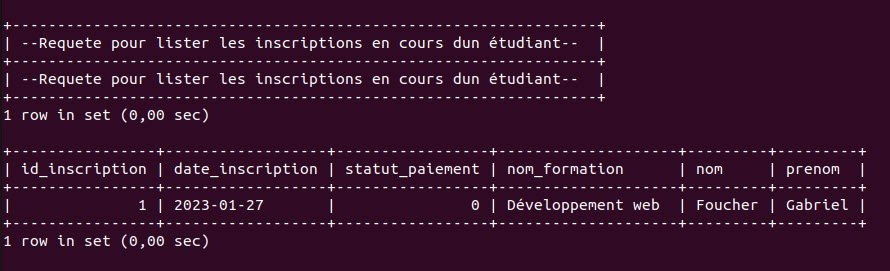
## Création des requêtes

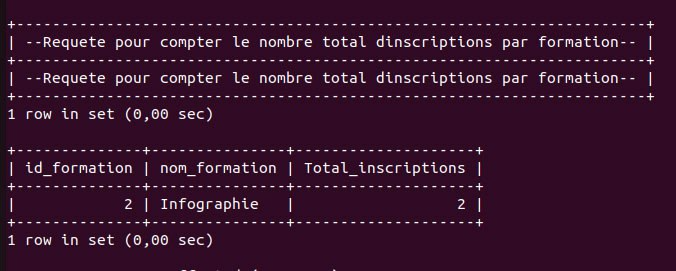


## Affichage des requêtes dans MySql

## 







## Définition des fonctions et procédures de notre système

## 

## Affichage de la fonction

## Mise en place des vues et triggers



## Affichage des vues et triggers dans MySql

# CONCLUSION

À travers notre base de données des inscriptions en ligne que nous avons créé, nous avons simplifié la collecte de données et amélioré l’efficacité de la gestion des inscriptions. Il n’est plus nécessaire de distribuer des formulaires papier à compléter et retourner.

Côté étudiants, tout est plus simple car ils peuvent s’inscrire quand et d’où ils veulent, depuis n’importe quel ordinateur ou téléphone connecté à Internet. Les informations fournies par les étudiants sont immédiatement stockées dans notre base de données.

Ce projet allège le travail des agents de la scolarité et augmente la visibilité de l’école.